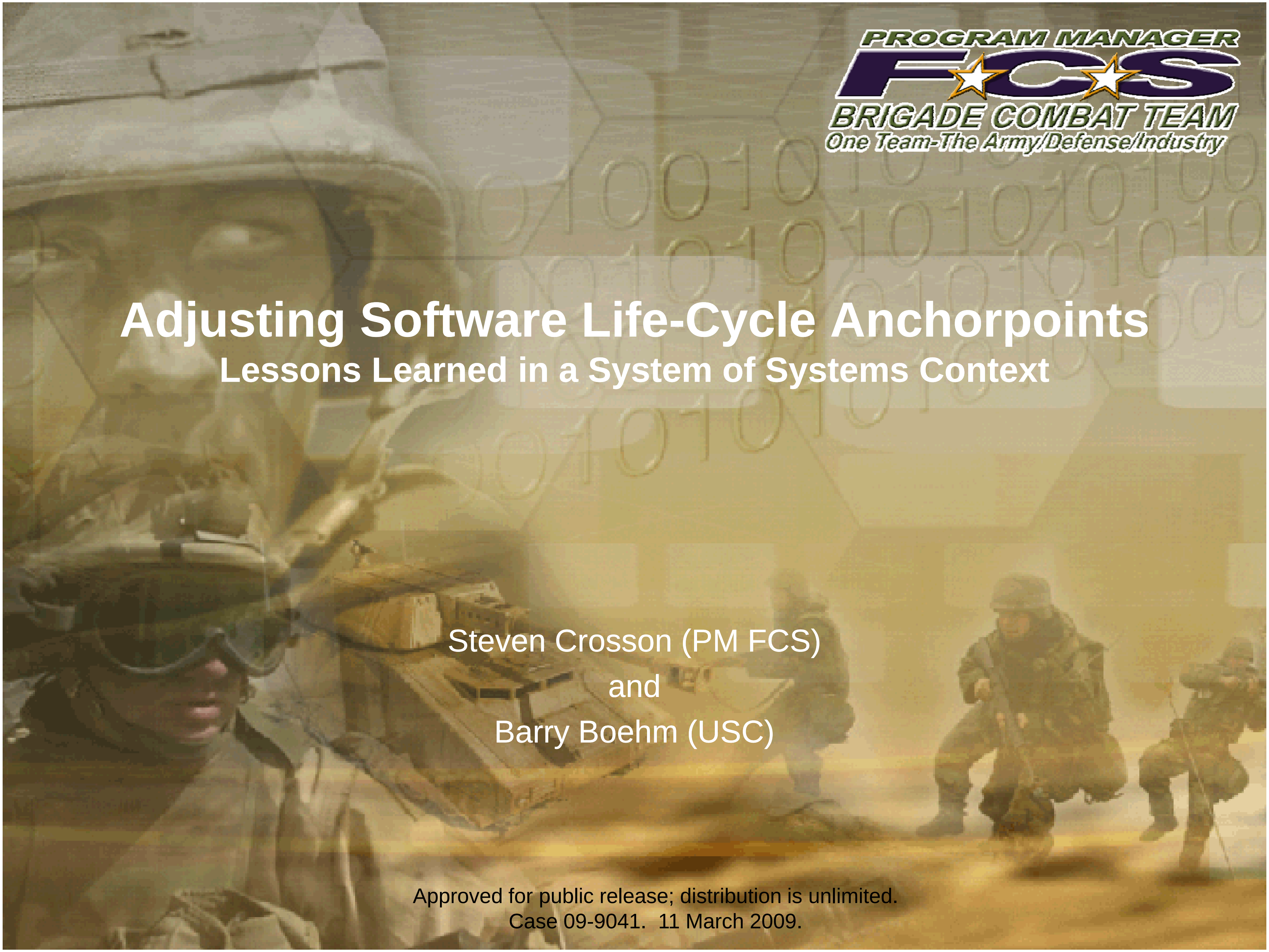


PROGRAM MANAGER FCS
BRIGADE COMBAT TEAM
One Team-The Army/Defense/Industry

Adjusting Software Life-Cycle Anchorpoints

Lessons Learned in a System of Systems Context

Steven Crosson (PM FCS)
and
Barry Boehm (USC)

Approved for public release; distribution is unlimited.
Case 09-9041. 11 March 2009.

Agenda

- Overview/Introduction
- Concept of the SoS LCA
- Establishing Focus Area Priority
- Results of Data Analysis
- Conclusions
- Epilogue

Introduction

Introduction – Software Review Basics

- Schedule and/or Event driven reviews are key to any major software development project

- Different reviews provide different benefits
  - Producibility
  - Capability
  - Integration and Test
  - Schedule

What is Future Combat Systems (FCS)?

- FCS is the Army’s modernization approach
- Utilizes a unique organizational structure utilizing the concept of a Lead System Integrator (LSI) to manage a collection of “best of industry” defense contractors
- Currently finishing a round of platform and network Preliminary Design Reviews (PDRs) and preparing for a System of Systems PDR (SoS PDR)
- Developing cutting-edge software and network functionality to provide increased capability to the Soldier

FCS Review Structure

- FCS Software is developed via several incremental builds
- Each build with its own set of software review events with specific foci:
  - Evaluation of functionality to be developed in software build against cost and schedule resources
  - Development of delivery, integration, and test timelines and criteria
  - Review of requirements and interface maturity
  - Horizontal and vertical integration of capabilities and requirements
- Each review is limited in scope to only apply to the current software build
- There was a need for the program to provide an overall assessment of the software development effort

Existing Software Review Events

- Two main levels of events are utilized for each FCS software build, one set for individual software packages and another for integrated software builds
- Individual software package reviews
  - Life Cycle Objective
  - Life Cycle Architecture
- Integrated software build reviews
  - Build Definition Checkpoint
  - Build Planning Checkpoint
  - Build Readiness Checkpoint
  - Build Assessment Checkpoint

Software Package Review Event Overviews

Event Name	Level of Review	Purpose of Review
Build Definition Checkpoint	Integrated software build	Present, discuss and agree on SoS level build plan and capabilities to be developed in this build.
Life Cycle Objective	Individual software package	Developers demonstrate understanding of the requirements and capabilities they need to provide in this build. Show understanding of the architecture and the ability to provide software within budget and schedule constraints.
Build Planning Checkpoint	Integrated software build	Roll-up of all artifacts presented at individual LCOs. Emphasis on horizontal integration of individual software package development.
Life Cycle Architecture	Individual software package	Focuses on design and prototype activities as well as analysis of the ability of software design to meet KPPs.
Build Readiness Checkpoint	Integrated software build	Considered the most important of checkpoint reviews. Serves as commitment , based on available evidence and data that capabilities for this build can be developed, tested, and delivered within existing budget and schedule limitations.
Build Assessment Checkpoint	Integrated software build	Focused on reviewing integration and test results and establishing lessons learned.

Need for an SoS LCA

- Existing events provide quality insight into the development plans and results of individual software builds

- Focus on evaluating design, architecture and requirements to develop a solution that *should* meet program needs

- However, very little emphasis on integrated quantitative modeling, simulation and test evidence reviews and evaluations to determine if the resultant software *is* going to meet program needs

- None of the events provide a review of the entire FCS software effort, but rather reviews of focused pieces of that effort

- There existed a need for an over-arching review of the results of the software development effort to date

Concept of the SoS LCA

Concept of the SoS LCA

- SoS LCA attempts to focus on test data and results (as well as a software producibility analysis) to evaluate current capability of the FCS software effort and project ability to meet program software capability needs

- This evaluation and forward projection of capability development ability of the FCS program will be a key feeder to the program SoS PDR event as well as Milestone C evaluation

- Event differs from existing FCS software reviews in that the SoS LCA focus is almost entirely on existing data and results, rather than plans

SoS LCA Areas of Review

PROGRAM MANAGER FCS BRIGADE COMBAT TEAM One Team-The Army/Defense/Industry

- Due to the size and scope of the FCS software development effort, narrowing down areas of review for the SoS LCA was a challenge in itself

- Tried to select areas (called “Focus Areas” by the review team) that would have impacts to the widest range of FCS IPT development efforts

- Focus areas selected were not necessarily risk areas, but just crucial to the successful development of the FCS software package

- Evaluation of each focus area wasn’t just on the status of that area, but the impact of that status on the entirety of the FCS software development effort

- Goal was to have a true System of Systems software evaluation

- If result of evaluation was less than current program plan, the analysis was to be accompanied by recommendations to resolve or mitigate any issues going forward, to provide the most capability possible to FCS

SoS LCA Focus Areas

Focus Area	Description
Software Producibility	Analysis of FCS software against cost and schedule models with acknowledgement that the scope and size of the FCS software effort is larger and more complex than most other projects.
End-State Design	Determination of whether the End State design will meet Operational needs with a focus on FCS KPPs.
Software Performance	Analysis of the processes for developing and testing software products, their effectiveness in testing requirements, and of the performance efficiency of the software.
Information Assurance and Security	Review of the status of efforts to develop and test FCS IA components, and a determination of the attainment of necessary functional capabilities
Distributed Information Management	Effectiveness of data communications between platforms and systems over the FCS network, analysis of data reliability, latency, etc
Distributed Fusion Management	Analyze effectiveness of utilizing several sources to gather and distribute fusion data, includes network usage, data reliability, latency, accuracy, etc.
Network Management	Review of current status of development of the Network Management system, as well as analysis of the current state of general network management issues.
Modeling & Simulation	Analysis of the process used to develop modeling and simulation test environments and supplemental functionality.

De Facto Review of Program Needs

- Due to SoS LCA review team resources, list of focus areas to be reviewed had to be narrowed down

- As a result of this, top-level analysis of the impacts to the FCS program of candidate focus areas were reviewed

- Result was a review of major software issues facing the program and an assessment of the level of impact they had on the software development effort

- Querying of all program IPTs for key software related issues provided not just focus areas for the SoS LCA, but an overall assessment and review of all program software review efforts

Resultant Lower Level Analysis

- While not chosen, these areas were reviewed in the process of selecting the SoS LCA focus areas and were part of the final review event analysis.

General Review Area	Comment
Software Interfaces	Via review of IA, security, and software performance, amongst others, necessitated a review of software interfaces. This included interface requirements and capabilities, as well as interface definition sessions and their effectiveness within the FCS software development effort.
Hardware Availability and Performance	While focus of the event was on software development, any software review on FCS must come with an accompanying understanding of hardware capability and delivery schedules. Tested and available hardware are necessary for any FCS operational software test.
Lessons Learned	Review of already completed test events provided ability to examine after-action reports and lessons learned from each event, providing additional insight on issues the SoS LCA could address
Integration Efforts	Through research to find available FCS test data came a review of how well FCS integration efforts were operating. This included having proper personnel and equipment as well as properly defined plans, processes and adequate time.

Establishing Focus Area Priority

Approved for public release; distribution is unlimited.
Case 09-9041. 11 March 2009.

Laying Foundation for Analysis

- In order to prepare for the analysis and presentation of eventual results, it was necessary to establish a hierarchy of focus area priority

- Result was that all focus area analyses would be rolled up as part, but not the entirety, of the Software Producibility and End-State Design focus areas

- The Producibility and End-State areas would provide the over-arching FCS software review from a budget and production standpoint and a technical standpoint respectively

Software Producibility Concept

- Unlike hardware, the software producibility costs for later increments tend to increase due to previous increment breakage as well as increased integration and test costs

- Using hardware-like estimates for future software increment producibility projections would lead to severe underestimation

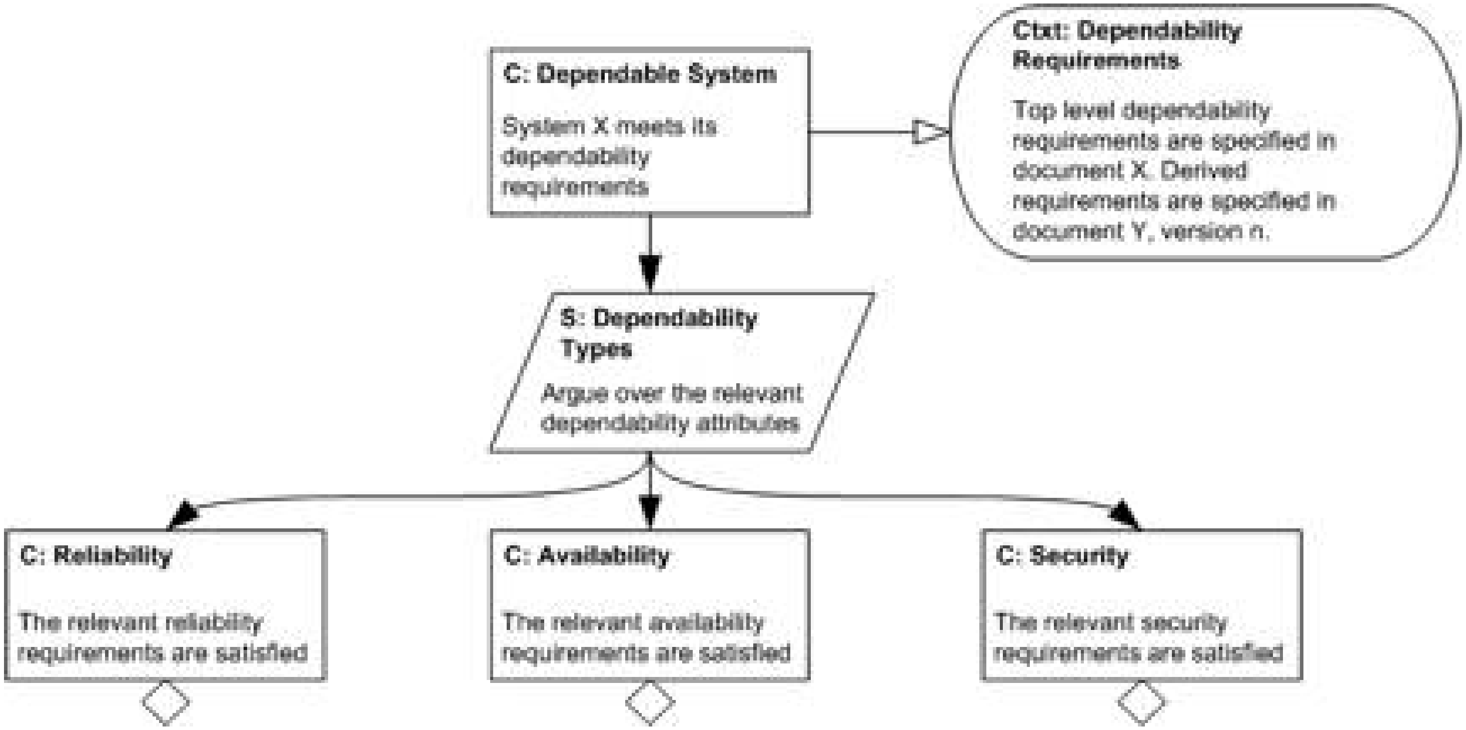
- To calibrate software estimates involves measurement of producibility data from early increment software deliveries

- The complexity and size of a program such as FCS further compounds the problem

End-State Design Concept

- Perhaps the most ambitious of the focus areas, as the depth of what this area attempted to address was unprecedented on the program
- Attempting to determine if the end state of FCS software at completion truly supports operational needs based upon currently available data, documentation and plans
- Decision was to base analysis around ability to realize all FCS KPPs
- Analysis was conducted through the use of Software Engineering Institute (SEI) assurance case methodology

Assurance Case Example


C: Dependable System
System X meets its dependability requirements
Ctxt: Dependability Requirements
Top level dependability requirements are specified in document X. Derived requirements are specified in document Y, version *n*.
S: Dependability Types
Argue over the relevant dependability attributes
C: Reliability
The relevant reliability requirements are satisfied
C: Availability
The relevant availability requirements are satisfied
C: Security
The relevant security requirements are satisfied

Courtesy of SEI: <http://www.sei.cmu.edu/pcs/acprep.html>

Approved for public release; distribution is unlimited.
Case 09-9041. 11 March 2009.

Results of Data Analysis

Preliminary Assessment

- Initial analysis of producibility and end-state focus areas (including the roll ups of other areas) indicated several areas of the FCS software development areas where key changes should be made
- Gaps in design, architecture and requirements coverage were identified
- A path to reach necessary capability in time for a Milestone C decision was identified
- Existing software reviews had specific artifacts and criteria to review, by being able to define appropriate and relevant criteria the SoS LCA gave the freedom to independently assess areas critical to FCS success

Conclusions

Value of the SoS LCA

- Both the depth and breadth of analysis of this software review event far exceeded other software-specific review events on the program
- The broad, multi-build SoS view, in conjunction with the individual system or individual build software reviews provide an excellent assessment of the current state of the FCS software development effort
- New areas of analysis provided insight into areas of the software development program that had never had an in-depth review
- Overall, the SoS LCA was a success both as a new piece to the FCS software review puzzle, but more importantly in providing a solid functional baseline for FCS software leading into SoS PDR

Epilogue

The Payoff

- Recommended changes to program processes and evaluations were proposed and are being developed

- Analyses not only helped discover problem areas, but also recognized software packages that were meeting or exceeding expectations
  - This latter category provided guidelines for recommended paths forward for other software packages

- Rather than mere action items to be tracked to answer specific questions, the SoS LCA provides new direction for certain program areas; more details and greater program benefit

References

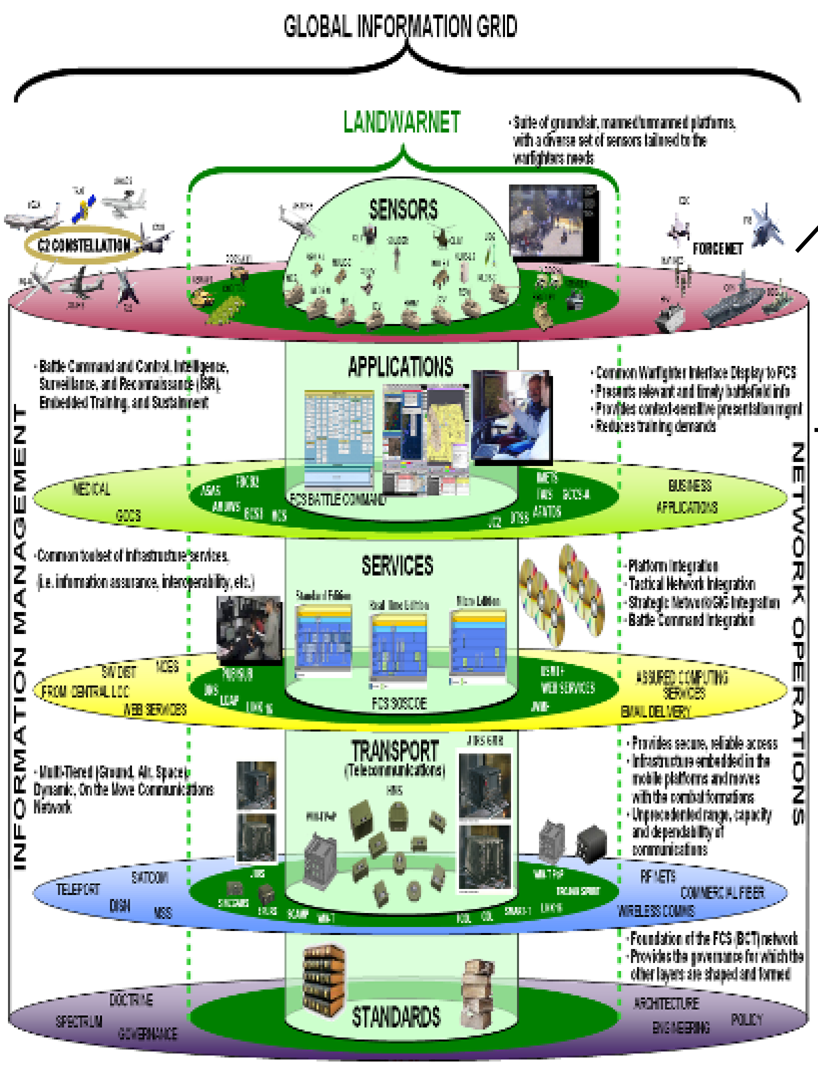
- [Boehm, 1996]. Barry Boehm, “Anchoring the Software Process”, IEEE Software, July 1996.
- [FCS SDP, 2007]. P. Goforth, et al, “Software Development Plan”, Future Combat Systems, June 2007.
- [FCS SoS LCA Plan, 2008]. A. Nguyen, et al, “Plan System of Systems Life Cycle Architecture Assessment”, Future Combat Systems, November 2008.
- [FCS SoS LCA Report, 2009]. A. Khan, et al, “Life Cycle Audit Assessment”, Future Combat Systems, March 2009.
- [SEI, 2007] C. Weinstock, “Assurance Case and Plan Preparation”, Software Engineering Institute, June 2007 <http://www.sei.cmu.edu/pcs/acprep.html>.

Backup

FCS Integrated Network

PROGRAM MANAGER FCS
BRIGADE COMBAT TEAM
One Team-The Army/Defense/Industry

FCS(BCT) Network



GLOBAL INFORMATION GRID
LANDWARNET
•Suite of ground/air, manned/unmanned platforms, with a diverse set of sensors tailored to the warfighters needs
C2 CONSTELLATION
FORCENET
SENSORS
APPLICATIONS
•Battle Command and Control, Intelligence, Surveillance, and Reconnaissance (ISR), Embedded Training, and Sustainment
•Common Warfighter Interface Display to FCS
•Presents relevant and timely battlefield info
•Provides context-sensitive presentation mgmt
•Reduces training demands
MEDICAL
GCCS
ASAS
FBCB2
AFATDS
BCS3
MCS
FCS BATTLE COMMAND
IMETS
TAIS
GCCS-A
AFATDS
JC2
DTSS
BUSINESS
APPLICATIONS
SERVICES
•Common toolset of infrastructure services,
(i.e. information assurance, interoperability, etc.)
Standard Edition
Real Time Edition
Micro Edition
•Platform Integration
•Tactical Network Integration
•Strategic Network/GIG Integration
•Battle Command Integration
SW DIST
NCES
FROM CENTRAL LOC
WEB SERVICES
PKI/SSUB
DNS
LDAP
LINK 16
FCS SOSCOE
JWNI
WEB SERVICES
JWP
ASSURED COMPUTING SERVICES
EMAIL DELIVERY
TRANSPORT
(Telecommunications)
JTRS GMR
HMS
•Multi-Tiered (Ground, Air, Space) Dynamic, On the Move Communications Network
•Provides secure, reliable access
•Infrastructure embedded in the mobile platforms and moves with the combat formations
•Unprecedented range, capacity and dependability of communications
TELEPORT
SATCOM
DISN
MSS
JTRS
SINCGARS
EPLRS
SCAMP
WIN-T
RF NETS
COMMERCIAL FIBER
WIRELESS COMMS
LINK16
•Foundation of the FCS (BCT) network
•Provides the governance for which the other layers are shaped and formed
STANDARDS
DOCTRINE
SPECTRUM
GOVERNANCE
ARCHITECTURE
ENGINEERING
POLICY
INFORMATION MANAGEMENT
NETWORK OPERATIONS

- Networked Sensors
- Commonality

Sensors

Battle Command

- Common Warfighter Interface
- Rapid Visualization of the Fight
- Shared Information/Planning/Execution
- Embedded Training and Logistics

SOSCOE

- Role Based Access
- Information Assurance
- Interoperability

JTRS GMR

JTRS HMS

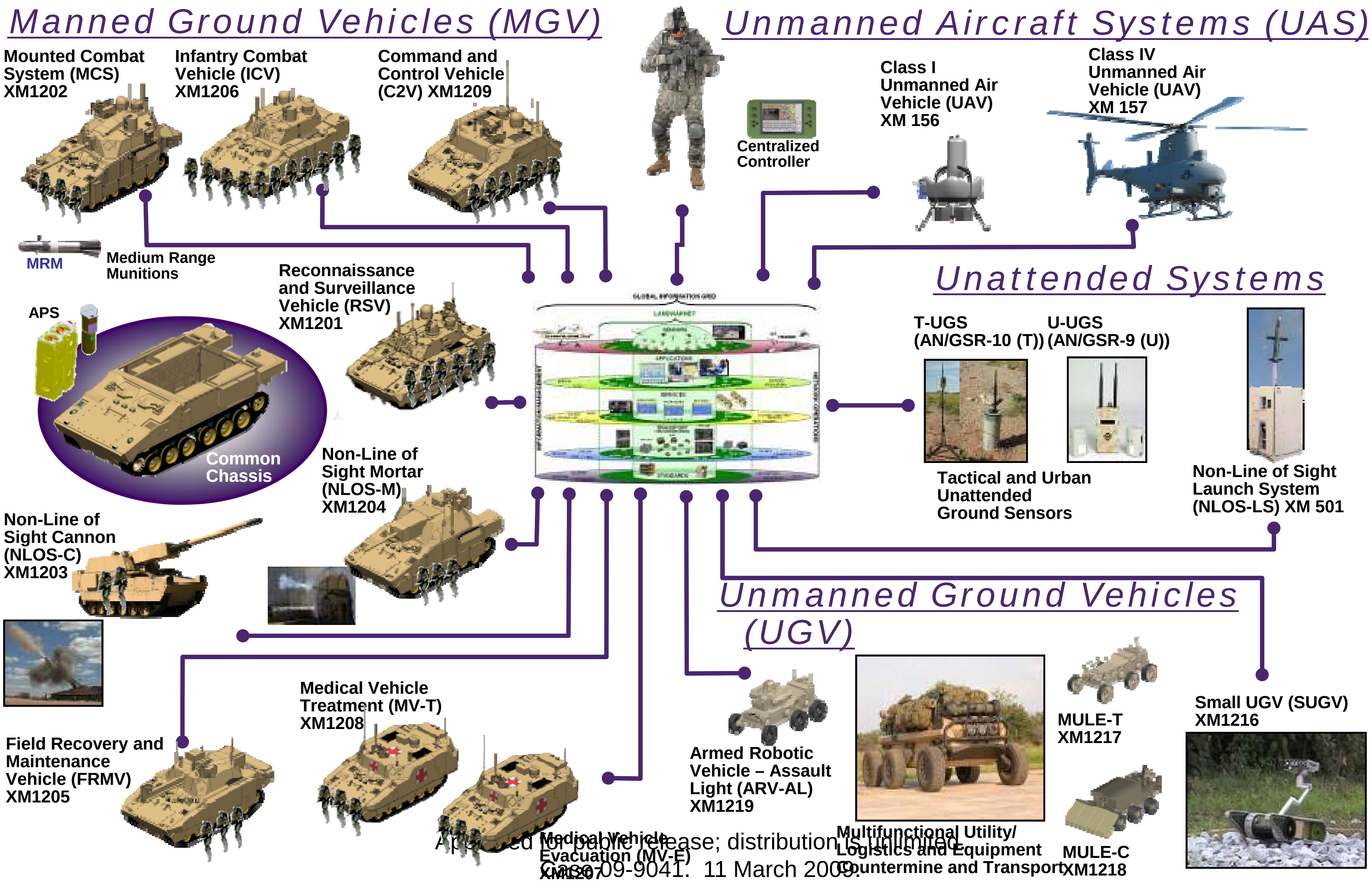
ICS

...lic relea... ...nlimited.
...09-9041. 11 March 2009.

- Network-enabled Battle Command on the Move
- Timeliness of Information To Squad Level in Seconds
- Units “Self-synchronize” as They Re-establish Network Connectivity

FCS Brigade Combat Team...

PROGRAM MANAGER FCS BRIGADE COMBAT TEAM
One Team-The Army/Defense/Industry



Manned Ground Vehicles (MGV)

Mounted Combat System (MCS) XM1202

Infantry Combat Vehicle (ICV) XM1206

Command and Control Vehicle (C2V) XM1209

MRM

Medium Range Munitions

APS

Common Chassis

Reconnaissance and Surveillance Vehicle (RSV) XM1201

Non-Line of Sight Mortar (NLOS-M) XM1204

Non-Line of Sight Cannon (NLOS-C) XM1203

Field Recovery and Maintenance Vehicle (FRMV) XM1205

Medical Vehicle Treatment (MV-T) XM1208

Medical Vehicle Evacuation (MV-E) XM1207

Unmanned Aircraft Systems (UAS)

Centralized Controller

Class I Unmanned Air Vehicle (UAV) XM 156

Class IV Unmanned Air Vehicle (UAV) XM 157

Unattended Systems

T-UGS (AN/GSR-10 (T))

U-UGS (AN/GSR-9 (U))

Tactical and Urban Unattended Ground Sensors

Non-Line of Sight Launch System (NLOS-LS) XM 501

Unmanned Ground Vehicles (UGV)

Armed Robotic Vehicle – Assault Light (ARV-AL) XM1219

Multifunctional Utility/ Logistics and Equipment Countermine and Transport

MULE-T XM1217

MULE-C XM1218

Small UGV (SUGV) XM1216

Approved for public release; distribution is unlimited. Case 09-9041. 11 March 2009.

Acronyms

- BAC – Build Assessment Checkpoint
- BCT – Brigade Combat Team
- BDC – Build Definition Checkpoint
- BPC – Build Planning Checkpoint
- BRC – Build Readiness Checkpoint
- FCS – Future Combat Systems
- IA – Information Assurance
- IEEE – Institute of Electrical and Electronics Engineers
- IPT – Integrated Product Team
- KPP – Key Performance Parameter
- LCA – Life Cycle Architecture
- LCO – Life Cycle Objective
- LSI – Lead System Integrator
- MGV – Manned Ground Vehicle
- MS C – Milestone C
- PDR – Preliminary Design Review
- PM – Program Manager
- SDP – Software Development Plan
- SEI – Software Engineering Institute
- SoS – System of Systems
- UAV – Unmanned Aerial Vehicle
- UGV – Unmanned Ground Vehicle
- USC – University of Southern California